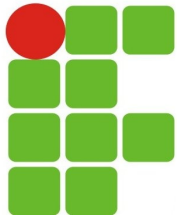
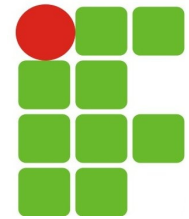


Sintaxe Básica do Java

João Paulo Q. dos Santos

joao.queiroz@ifrn.edu.br



Plano de Aula

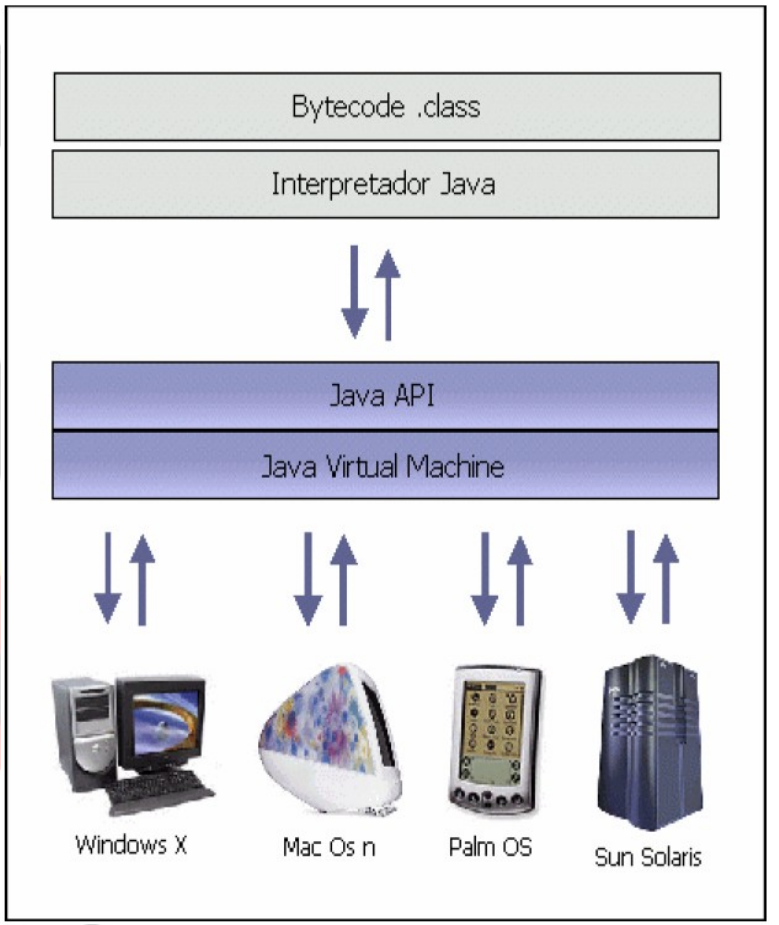
- Introdução;
- Método Main;
- Variáveis:
  - Tipos primitivos;
  - Referências para Objetos;
- Operadores;
- Estruturas: Condicionais e Repetição;
- Ler e Escrever;

Revisando

Java têm a proposta de "write once, run anywhere" (programe uma vez, e execute em qualquer lugar) a linguagem.

Como lidar com os diferentes sistemas operacionais (SO), e diversos ambientes computacionais existentes ?

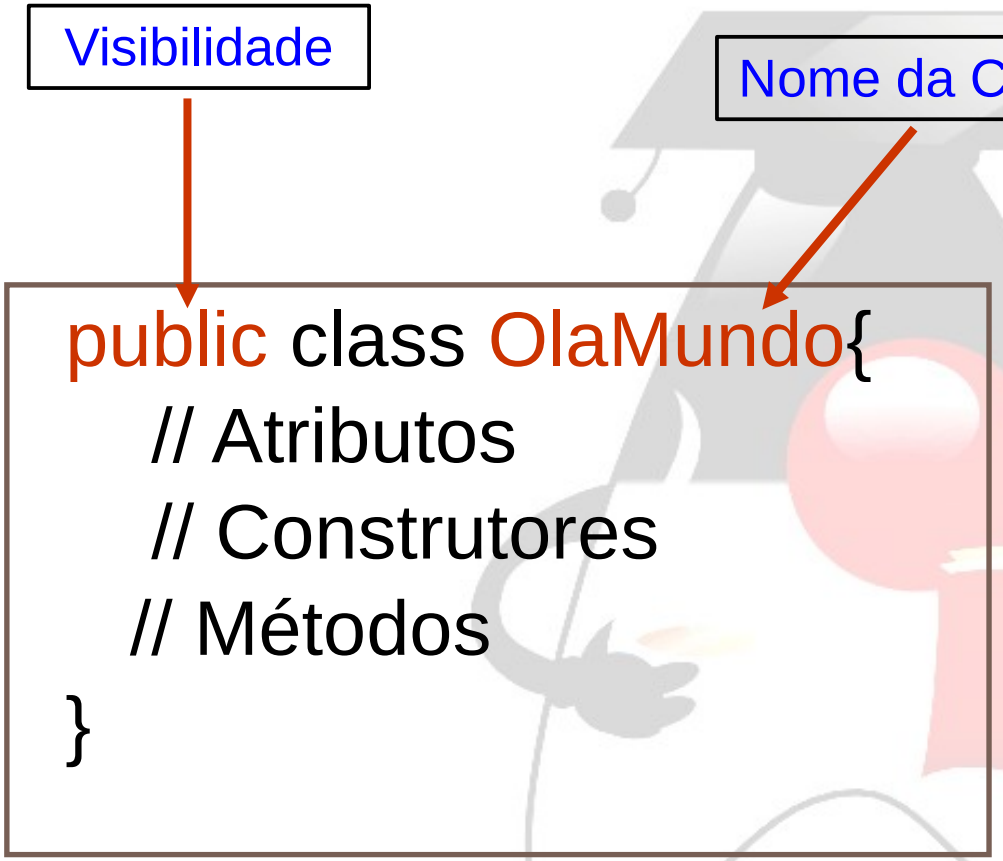
Surgiu a idéia de possuir uma camada intermediária entre o SO e o programa em java a ser executado, esta camada chama-se **Java Virtual Machine (JVM).**



Classe OlaMundo

Método **main** em Java

Uma classe em Java



Obs:

O nome da classe deve ser **EXATAMENTE** igual ao do Arquivo.

Um arquivo só pode conter exatamente uma classe.
EXCETO para as Inner Class.

```
public class OlaMundo{
    public static void main (String args[]){
        System.out.println("Olá Mundo");
    }
}
```

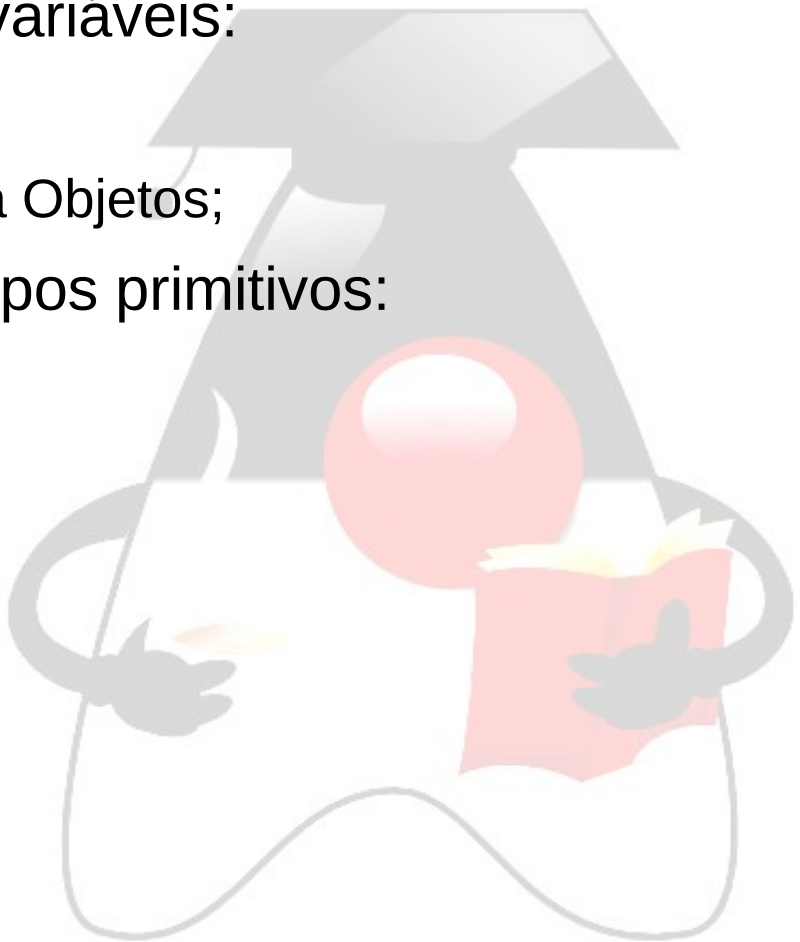
Informa que este método pode ser executado sem que precise instanciar um objeto.

Este método torna a classe executável.

Método Especial

Variáveis

- Duas classes de variáveis:
  - Tipos primitivos;
  - Referências para Objetos;
- Java possui oito tipos primitivos:
  - byte,
  - short,
  - int,
  - long,
  - float,
  - double,
  - char,
  - boolean;



Tipos Primitivos

Palavra	Descrição	Tamanho/formato
	Inteiros	
byte	Inteiro de um byte	8 bits
short	Inteiro pequeno	16 bits
Inteiros	Inteiro	32 bits
long	Inteiro longo	64 bits
	Números reais	
float	Ponto flutuante de precisão simples	32 bits
double	Ponto flutuante de precisão dupla	64 bits
	Outros tipos	
char	Um caractere	16 bits – Unicode
boolean	Um valor logico	true ou false

Declaração de Variável

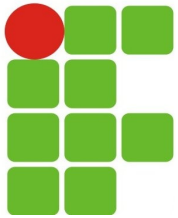
<Tipo> <Identificação>

Exemplo

int numero;

Nota:

- **Variáveis ficam na pilha e armazenam valor;**
- **Possuem tamanho fixo.**



Literais

- **Valores usados diretamente no código**
  - Possuem, implicitamente, um tipo
- **Inteiros**
  - 10 ; 832 ; 2 ; -832
- **Ponto flutuante**
  - 3.2 ; 4.3 ; 1232.1232 ; 32.2 ; 3.2f
- **Caractere**
  - 'a' ; '\u0041' ; '\u0065'
- **Booleano**
  - true ; false

Nome de variáveis

- **Nomes de Variáveis:**
  - Podem conter caracteres, dígitos, _ e $**;**
  - Devem começar com caractere, _ ou $;
  - **Não** podem conter espaços;
  - Maiúsculas **diferentes** de minúsculas;
  - É convencional (mas não obrigatório) usar uma **letra minúscula** para a **primeira letra** do nome de uma variável;
  - Todas as variáveis de **tipos primitivos** precisam ser **declaradas** e **inicializadas;**
- **Exemplos:**
  - nota ; x ; y ; raio ; mediaTotal
  - media_total ; media$total ; media1
  - media2 ; media_1 ; media$1
  - nomePai ; nome_pai ; nome_mae

Coersão *(casting)*

- Um número pode sempre ser atribuído a outro de maior precisão:

```
int a = 274;
long b = a;
```

- A operação inversa requer ***coersão*** explicita (*"casting"*):

```
long a = 274;
int b = (int) a;
```

Referências à objetos

As variáveis associadas a objetos, que são referências aos mesmos, precisam ser declaradas. Os objetos por sua vez precisam ser criados. Declaração e criação podem sem realizadas por comandos separados:

Sintaxe:

```
<NomeDaClasse> <nomeDoObjeto>;

<nomeDoObjeto> = new <NomeDaClasse()>;
```

Exemplos:

```
Aluno aluno;

aluno = new Aluno();
```

```
Aluno aluno = new Aluno();
```

Referências à objetos

Os critérios para nomear um objeto são os mesmos para nomear uma variável.

Operadores Aritméticos e de Atribuição

Operador	Significado	Operador	Exemplo	Expressão equivalente
+	adição	+=	x += y	x = x + y
-	subtração	-=	x -= y	x = x - y
*	multiplicação	*=	x *= y	x = x * y
/	divisão	/=	x /= y	x = x / y
%	resto da divisão (módulo)	%=	x %= y	x = x % y

Atribuição

- Os operadores possuem uma forma de atribuição:

  Variável = Expressão;

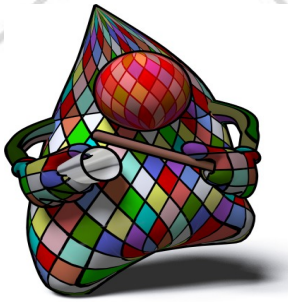
- Exemplo:

  int a = 10;

  a += 10; é o mesmo que a = a + 10;

Operadores de incremento e decremento

Operador	Exemplo	Significado
++	++a	O incremento será feito antes ao uso do valor da mesma na expressão
	a++	O incremento será feito após o uso do valor da mesma na expressão
--	--a	O decremento será feito antes ao uso do valor da mesma na expressão
	a--	O decremento será feito após o uso do valor da mesma na expressão



Exemplo:
Pós-Incremento e Pré-Incremento

```
public class Incremento {
    public static void main(String[] args) {
        int a=0;//inicializando a variável a
        if ((a++ + a++)==1){
            System.out.println(a);
        }
        System.out.println(a++);
        System.out.println(a);
    }
}
```

Qual o valor de a ?

Qual o valor de a ?

Qual o valor de a ?

Operadores de comparação ou relacionais

Operador	Significado
==	Igual a
!=	Diferente de
<	Menor que
>	Maior que
<=	Menor ou Igual a
>=	Maior ou Igual a

Estes operadores atuam sobre valores numéricos e retornam valores booleanos, **true** (verdadeiro) ou **false** (falso)

Operadores lógicos

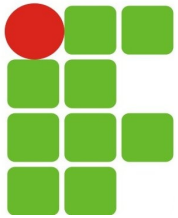
Operador	Significado
&&	E lógico (*"logical AND"*)
&	E lógico boleano(*"boolean logical AND"*)
||	OU logico (*"logical OR"*)
|	OU lógico Inclusivo(*"boolean logical inclusive OR"*)
^	OU EXCLUSIVO exclusivo (*"boolean logical exclusive OR"*)
!	NÃO lógico(*"logical NOT"*)

Exemplo: Operadores Lógicos

Exemplo	Explicação
a && b	retorna **true** se **a** e **b** forem ambos **true**. Senão retorna **false**. Se **a** for **false**, **b** não é avaliada **(Curto Circuito)**.
a & b	retorna **true** se **a** e **b** forem ambos **true**. Senão retorna **false**. Ambas expressões **a** e **b** são **sempre avaliadas**.
a || b	retorna **true** se **a** ou **b** for **true**. Senão retorna **false**. Se **a** for **true**, **b** não é avaliada **(Curto Circuito)**.
a | b	retorna **true** se **a** ou **b** for **true**. Senão retorna **false**. Ambas expressões **a** e **b** são **sempre avaliadas**.
a ^ b	retorna **true** se **a** for **true** e **b** for **false** ou vice-versa. Senão retorna **false. Apenas um pode ser true**
!a	retorna **true** se **a** for **false**. Senão retorna **false**

Precedência operadores

Operador	Descrição
. [ ] ( ) (tipo)	Máxima precedência: separador, indexação, parâmetros, conversão de tipo
+ - ~ ! ++ --	Operador unário: positivo, negativo, negação (inversão bit a bit), não (lógico), incremento, decremento
* / %	Multiplicação, divisão e módulo (inteiros)
+ -	Adição, subtração
<< >> >>>	Translação (bit a bit) à esquerda, direita sinalizada, e direita não sinalizada (o bit de sinal será 0)
< <= >= >	Operador relacional: menor, menor ou igual, maior ou igual, maior
== !=	Igualdade: igual, diferente
&	Operador lógico e bit a bit
^	Ou exclusivo (xor) bit a bit
|	Operador lógico ou bit a bit
&&	Operador lógico e condicional
||	Operador lógico ou condicional
?:	Condicional: if-then-else compacto
=	Atribuição



Exemplo 1: Curto Circuito

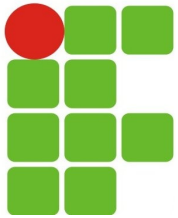
```
public class CurtoCircuito {
    public static void main(String[] args) {

        int a = 10;
        int b = 0;

        if (a <= 10 || (++b) == 0) {
            System.out.println(b);
        }
    }
}
```

Qual o valor de b?

Exemplo 2: Curto Circuito

```
public class CurtoCircuito {
    public static void main(String[] args) {

        int a = 10;
        int b = 0;

        if (a > 10 && (++b) == 0) {
            System.out.println(“caso 1”);
            System.out.println(b);

        } else {
            System.out.println(b);
            System.out.println(“caso 2”);
        }
    }
}
```

Qual o valor de b?



Exemplo

- Compile o programa abaixo e execute:
  - java NomeClasse par1 par2 par3

```
public class Exemplo{

    public static void main(String[] args) {

        System.out.print("Quantidade de parâmetros: ");
        System.out.println(args.length);

        for (int i = 0 ; i < args.length ; i++) {

            System.out.println(args[i]+" ");
        }
    }
}
```

Estrutura condicional

```
if (<condicional>) {
   <Comando>;
   <Comando>;
}
```

```
if <condicional>
     <Comando>;
```

```
If( <condicional>){
     <Comando>;
 }else if (<condicional>) {
     <Comando>;
 }
```

Exemplo
if / else / if else

```
if (idade > 17){

     System.out.println("Maior de Idade");

}else{
     System.out.println("Menor de Idade");
}
```

```
if ((hora > 4) && (hora < 13)){

     System.out.println("Bom Dia");

}else if ((hora >12) && (hora < 19)){

     System.out.println("Boa Tarde");

} else{
     System.out.println("Boa Noite");
}
```

Estrutura condicional

```
switch(expressão){
    case valor1:
        instruções;
      break;
    case valor2:
      instruções;
      break;
     default:
      instruções;
}
```

Expressão deve ser uma Uma variável de tipo igual a int ou char.

Exemplo swicth

```
switch(menu){

      case 1:
         System.out.println("Sopa");
         break;

      case 2:
         System.out.println("Bife");
         break;

      default:
       System.out.println("Água");
}
```

Expressão deve ser uma Uma variável de tipo igual a int ou char.

Estrutura de repetição

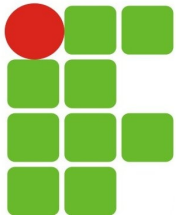
Entra no laço enquanto a condição for verdadeira.

Só entra no laço se a condição for verdadeira. Teste no início.

```
while <condição> {
   <comandos>
}
```

Entra no laço no mínimo uma vez. Teste no final.

```
do{
   <comandos>
}while <condição>;
```



Exemplo while / do while

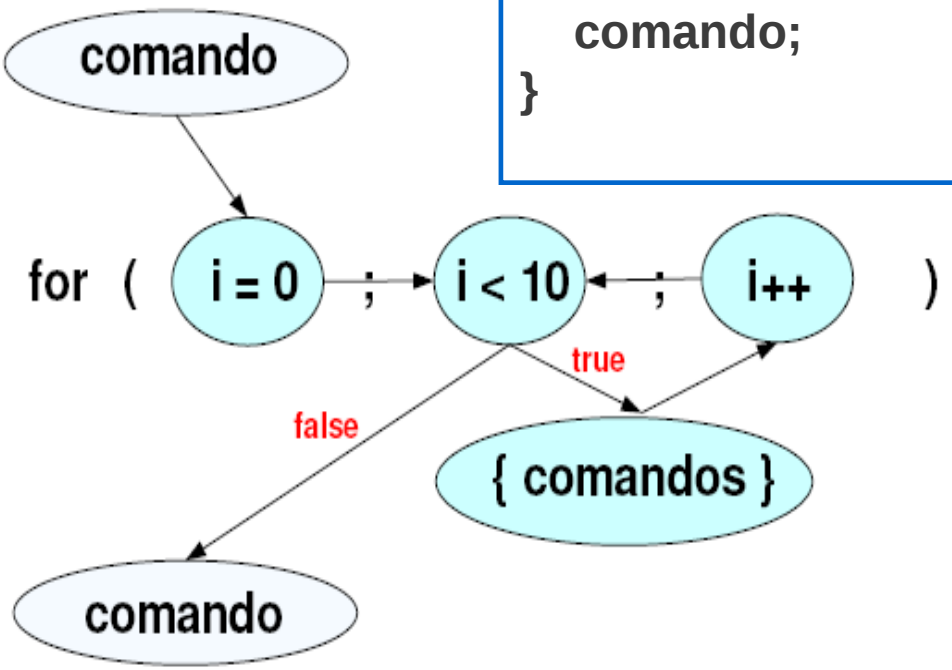
```
public class While {

   public static void main(String[] args) {

       int i = 0;

       while (i < 10) {
          System.out.println(i++);
       }

       do {
          System.out.println(i--);
       } while (i >= 0);
    }
}
```

Estrutura de repetição

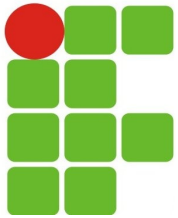
A condição de parada é previamente conhecida.

Sintaxe

```
for (inicializações ; condição de fim ; passo ){
    comando;
}
```

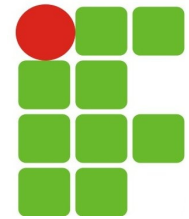


Execução

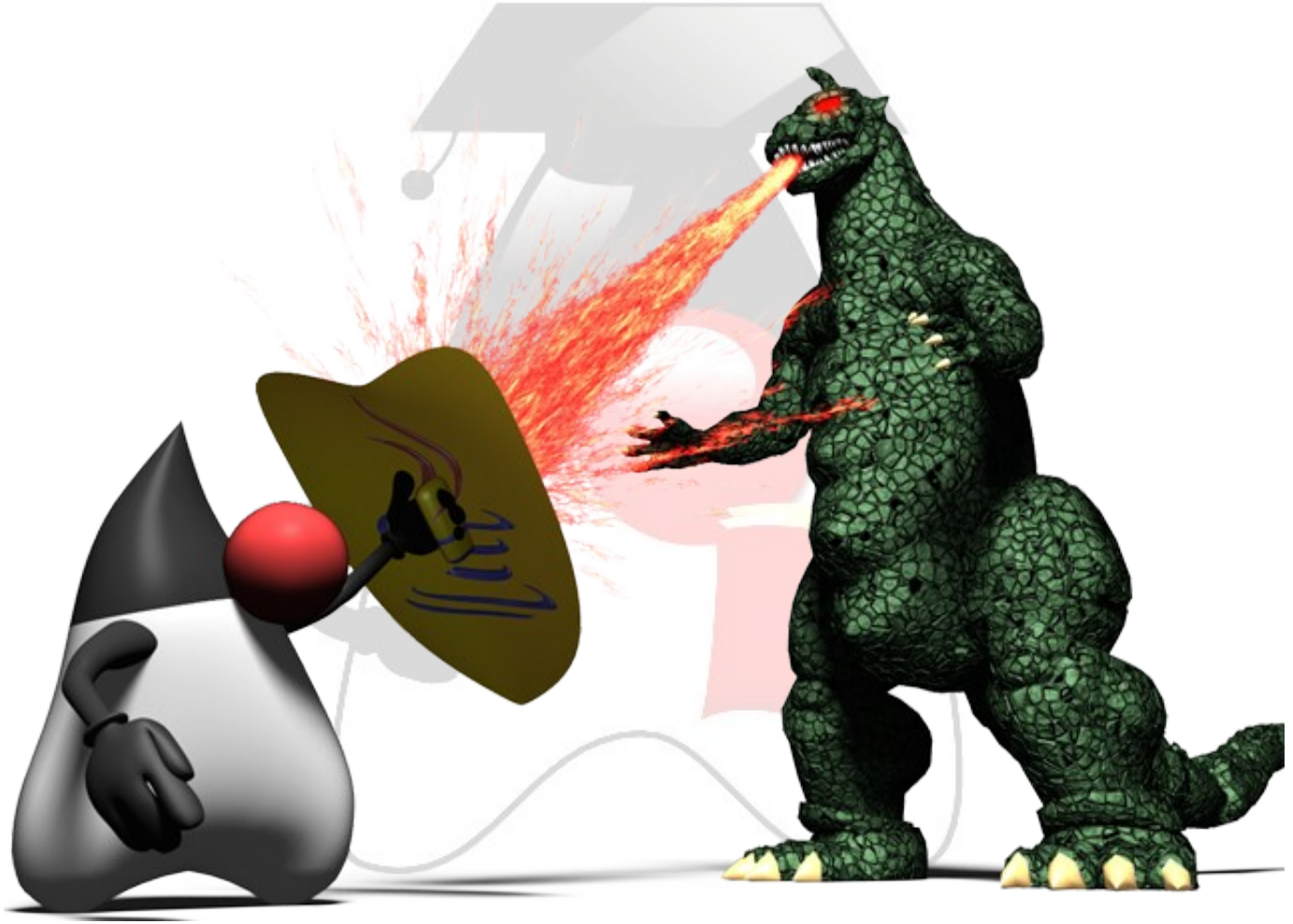


Exemplo for

```
public class For {

    public static void main(String[] args) {

        int condicaoParada = 1000;

        for (int i = 1; i <= condicaoParada; i++){
            if (i%2==0)
              System.out.println(i+" : é Par;");
            else
               System.out.println(i+" : é Impar;");
        }
    }
}
```



Ler e Escrever

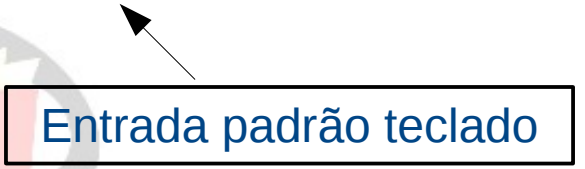


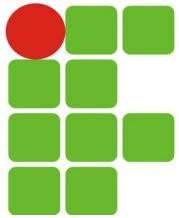
Escrever no console

- Atributo out da classe System
  - metodo print e println;
- Exemplos:
  - System.out.println(“Teste”);
    - Escreve Teste e avança uma linha
  - System.out.print(“Teste”);
    - Escreve Teste e não avança linha
- Pode-se usar o **“+” p/ concatenar:**
  - System.out.println(“Numero:”+n);

Ler do teclado

- Necessário criar objeto da classe Scanner:
  - Definir uma variável do tipo Scanner;
  - Importar a classe java.util.Scanner;
- Criar um objeto para ler do teclado:
  - Scanner sc = new java.util.Scanner(**System.in**);
- Ler dados:
  - int x = sc.nextInt();// Ler um inteiro





Ler do teclado

- **Tipos primitivos:**
  - nextByte()
  - nextShort()
  - nextInt()
  - nextLong()
  - nextFloat()
  - nextDouble()
  - nextBoolean()
- **Objetos:**
  - next();
  - nextLine()
  - nextBigDecimal()
  - nextBigInteger()

Exemplo:
Ler e escrever

```
import java.util.Scanner;
public class LerTeclado {

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

     System.out.println("Digite um valor para A:");

      int a =  sc.nextInt();

     System.out.println("Digite um valor para B:");

      int b =  sc.nextInt();

      System.out.println("A Soma de A + B = "+(a+b));
  }
}
```

Importa a classe Scanner que está no pacote java.util

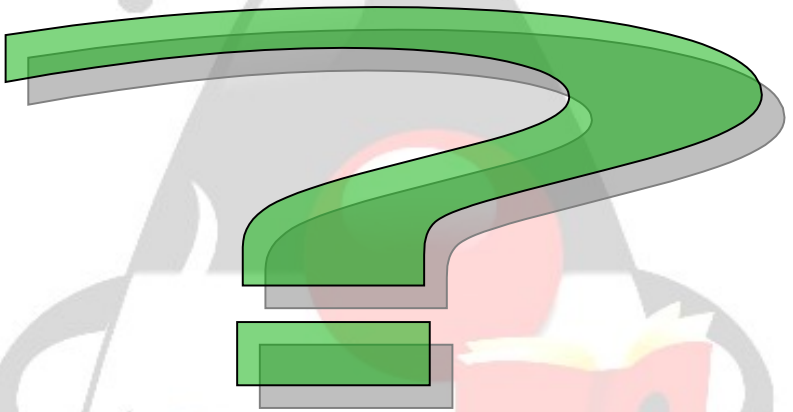
Cria um objeto do tipo

Ler um valor inteiro do teclado

Ler um valor inteiro do teclado

Escreve no console o resultado da soma

Dúvidas



Exercício de fixação

1 - Fazer uma calculadora que solicita dois valores do usuário e exibe na Tela os seguintes resultados:

- Soma;
- Subtração;
- Divisão e
- Multiplicação.

2 - Fazer uma aplicação que solicita do usuário o:

- nome;
- idade;
- e-mail;

Após receber estes parâmetros, o sistema deve exibir na tela o seguinte:

- Se o usuário tiver mais de 18 anos: Cadastro realizado com sucesso
- Se o usuário tiver menos de 18 anos: Cadastro na fila de espera;
- Se o usuário for maior de 60 anos: Cadastro registrado no passe livre.

Exercício de fixação

3 – Fazer uma aplicação que solicita do aluno três notas. O sistema deve realizar a média entre as três notas e informar se o aluno foi:

- Aprovado: média >= 7;

- Recuperação: média > 4 e média < 7;

- Reprovado: média < 5.

OBS: A média é um valor inteiro.